ELECTROMAGNETIC & SENSOR SYSTEMS
DEPARTMENT

Identifying Aliases in Graphs

David J. Marchette

May 21, 2008

Quantitative Methods in Defense and National Security

Report Documentation Page	*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE **21 MAY 2008**	2. REPORT TYPE	3. DATES COVERED **00-00-2008 to 00-00-2008**

4. TITLE AND SUBTITLE **Identifying Aliases in Graphs**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Naval Surface Warfare Center,Electromagnetic & Sensor Systems Department,Dahlgren,VA,22448**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR’S ACRONYM(S)
	11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

Presented at the Quantitative Methods in Defense and National Security (QMDNS), 21 May 2008, Durham NC

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**27**	

Outline

Motivation

Definitions and Model

Alias Identification

Conclusions

Outline

Motivation

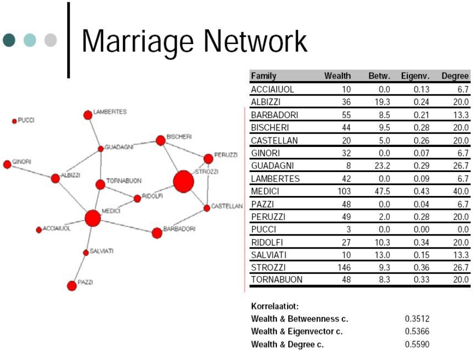
Definitions and Model

Alias Identification

Conclusions

Social Networks

- A model of the relationships between entities.
- Also used to study insurgent groups, terrorist cells, etc.
- Relates actors (nodes in the network) through relationships (edges in the network).
- Typically used for small groups, with full knowledge of all links.



Marriage Network

Family	Wealth	Betw.	Eigenv.	Degree
ACCIAIUOL	10	0.0	0.13	6.7
ALBIZZI	36	19.3	0.24	20.0
BARBADORI	55	8.5	0.21	13.3
BISCHERI	44	9.5	0.28	20.0
CASTELLAN	20	5.0	0.26	20.0
GINORI	32	0.0	0.07	6.7
GUADAGNI	8	23.2	0.29	26.7
LAMBERTES	42	0.0	0.09	6.7
MEDICI	103	47.5	0.43	40.0
PAZZI	48	0.0	0.04	6.7
PERUZZI	49	2.0	0.28	20.0
PUCCI	3	0.0	0.00	0.0
RIDOLFI	27	10.3	0.34	20.0
SALVIATI	10	13.0	0.15	13.3
STROZZI	146	9.3	0.36	26.7
TORNABUON	48	8.3	0.33	20.0

Korrelaatiot:

Wealth & Betweenness c.	0.3512
Wealth & Eigenvector c.	0.5366
Wealth & Degree c.	0.5590

Covert Networks

- Actors have a vested interest in not being observed.
- Networks may be very large.
- The networks change in time.
- Some links are known to be there, some known to be missing, but others are unknown.
- An actor may try to hide (change email address, change phone number, start calling themselves Colonel Guapa).

Methodology

- Assume the existence of a “social space” $\mathcal{S}$ which controls the structure of the network.
- The probability of an edge in the network is a function of the “closeness” of the nodes in $\mathcal{S}$.
- The social space provides a framework from which inference can be performed.

Social Space

- Early work reported by Hoff et al in JASA.
- Model based on location:
  - Probability of an edge between v_i and v_j a function of their distance in social space.
  - Several variations proposed.
- Versions of the Exponential Random Graph Models (ERGMs) (Hunter et al, JASA 2008) can be thought of in terms of a “social space”.
- We will discuss a “social space” model that has a simple least squares algorithm for fitting the parameters, which can be used on large graphs (thousands to tens of thousands of nodes or more).

Outline

Motivation

Definitions and Model

Alias Identification

Conclusions

Graph Definitions

- A graph is a pair (V, E) where V is a set (vertices) and E is a collection of unordered pairs of vertices (edges).
- We can consider directed graphs (V, A) where A (arcs or arrows) are ordered pairs.
- The order of the graph is $|V|$ and the size of the graph is $|E|$ (or $|A|$ in the case of directed graphs (digraphs)).
- Vertices are sometimes called “nodes” or “actors”.
- Edges are sometimes called “links” or “relations”.
- The adjacency matrix $A = (a_{ij})$ is the $|V| \times |V|$ binary matrix with a 1 in those places where an edge occurs in the graph.

Probabilistic Framework

- We place a probability structure on the network.
- This means we fit a **generative** model to the graph.
- This allows us to estimate the probability of a missing (unknown) link.
- We can bring node attributes into the model.
- We are essentially choosing the “most likely” graph given the model assumption and the observed edges.

Random Dot Product Graphs

- Each vertex v_i has associated with it a vector x_i.
- Place an edge $v_i v_j$ between vertices v_i and v_j with probability proportional to $x_i x_j$, the dot product of x_i and x_j.
- Thus $p_{ij} = f(x_i x_j)$. We'll use the threshold function for f:

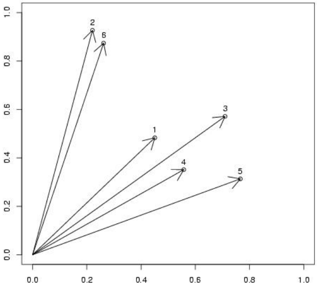
$$f(x) = \begin{cases} 0 & x < 0 \\ x & 0 \leq x \leq 1 \\ 1 & x > 1 \end{cases}$$

- The edges in the random graph are no longer independent.
- We need to estimate the x_i from the observed graph.
- We can extend the model to directed graphs by having in- and out-vectors x_i^I and x_i^O with p_{ij} proportional to $x_i^O x_j^I$.

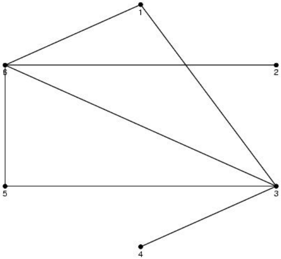
$\mathcal{S}$

- Each vertex v_i has associated with it a vector $x_i \in \mathcal{S}$.
- The proximity (as measured by the dot product) of two vectors controls the probability of an edge.
- Thus $\mathcal{S}$ is the space which defines the random graph that we observe.

$\mathcal{S}$



$\mathcal{G}$



Linear Algebra (Least Squares)

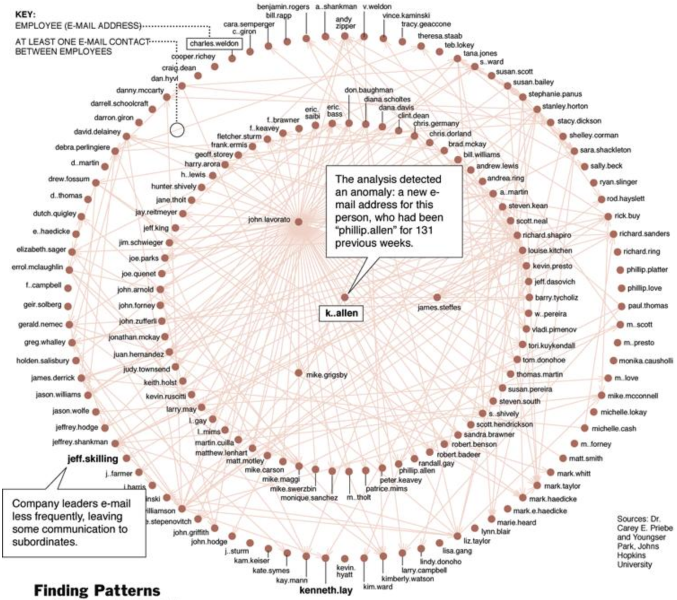
Note that if we want to find the vectors U which best “match” the adjacency matrix A (best in Frobenius norm), then the singular value decomposition: $A = UDV'$ almost works (the problem is the diagonal). Note that for graphs A is symmetric, so $V = U$.

1. Set $D = \text{diag}(0)$.
    - 1.1 $s = \text{svd}(A + D)$.
    - 1.2 $X = s\$U$, scaled by the singular values.
    - 1.3 $D = \text{diag}(XX')$.
2. Repeat 1–3 until convergence.
3. Return X.

The Enron Data

- Graphs (directed graphs) of emails between executives at Enron.
- 184 email addresses (nodes).
- 150 executives (names).
- 187 weeks.
- Each graph corresponds to 1 week of emails.
- An edge $v \rightarrow w$ if there was an email from v to w within the week.
- Note: we are ignoring multiple emails and an email from one to many generates a “star” of edges.

An Alias



The analysis detected an anomaly: a new e-mail address for this person, who had been "phillip.allen" for 131 previous weeks.

Company leaders e-mail less frequently, leaving some communication to subordinates.

Sources: Dr. Carey E. Priebe and Youngser Park, Johns Hopkins University

Finding Patterns In Corporate Chatter

Computer scientists are analyzing about a half million Enron e-mails. Here is a map of a week's e-mail patterns in May 2001, when a new name suddenly appeared. Scientists found that this week's pattern differed greatly from others, suggesting different conversations were taking place that might interest investigators. Next step: word analysis of these messages

The Alias

- k..allen did not appear in any prior graph.
- Perusal of the content of the emails determines that these were sent by Phillip Allen.
- phillip.allen appears in the previous graphs.
- A matched filter comparing neighborhoods was implemented and it found the correct match.
- In this work, we develop a “social space” version of the matched filter.

Outline

Motivation

Definitions and Model

Alias Identification

Conclusions

Aliases

- Given two graphs G_t and G_{t+1}.
- Suppose we know some of the vertices are shared by these graphs (and which ones they are).
- There is one vertex in G_{t+1} that we have not seen before.
- Assuming that this vertex appeared in G_t with a different label, can we determine this vertex?

Aliases

- Setup:
  - Two graphs, $G_t = (V \cup U_t, E_t)$ and $G_{t+1} = (V \cup U_{t+1}, E_{t+1})$.
  - All vertices are labeled (email addresses).
  - Vertices in V are named (individual associated with the address).
  - Vertices in U_i are not named.
- Want to associate the names to the vertices in U_{t+1}.

Methodology

- Assign the name to vertex u whose vector x_v is closest to the vector x_u.
- Optimize:

$$(X, Y_1, Y_2) = \arg\min_{X,Y_1,Y_2} \left\| \left(\begin{pmatrix} X \\ Y_1 \end{pmatrix} \begin{pmatrix} X \\ Y_1 \end{pmatrix}^T \right)_0 - A_1 \right\|_F + \left\| \left(\begin{pmatrix} X \\ Y_2 \end{pmatrix} \begin{pmatrix} X \\ Y_2 \end{pmatrix}^T \right)_0 - A_2 \right\|_F ,$$

- M_O means M with the diagonal replaced with zeros.
- Thus, we are attempting to fit a set of vectors to the known and a set each for the unknown in the two graphs. Fitting to the knowns constrains the Y_i to lie in the same space.

The Setup

- Input A_1, A_2, the adjacency matrices of the graphs corresponding to the vertices (V, U_i).
- Set B to be the average of $A_1[V]$ and $A_2[V]$, the blocks corresponding to V.
- Set $N = n + n_1 + n_2$.
- Set A to be the $N \times N$ matrix with first $n \times n$ block equal to B, and blocks $A[V, U_i] = A_i$, $A[U_i, V] = A_i'$.

$$A = \begin{pmatrix} \frac{A_1[V,V]+A_2[V,V]}{2} & A_1[V, U_1] & A_2[V, U_2] \\ A_1[U_1, V] & A_1[U_1, U_1] & Y' \\ A_2[U_2, V] & Y & A_2[U_2, U_2] \end{pmatrix}$$

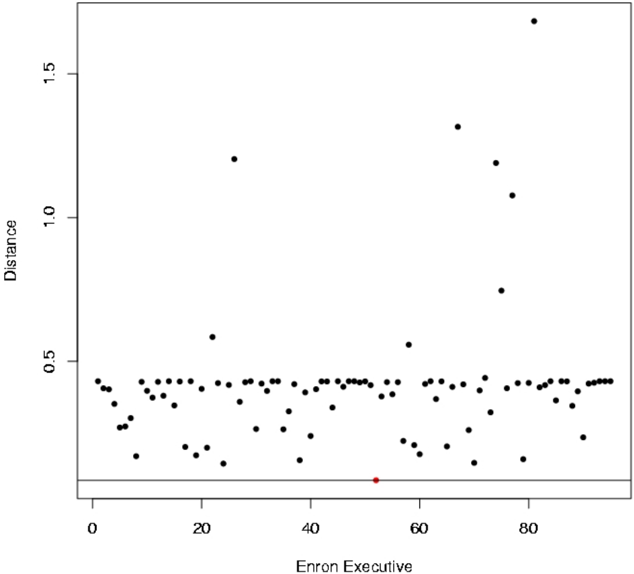
where Y is the dot product of vectors derived from U_1 and U_2.

Fitting the Alias

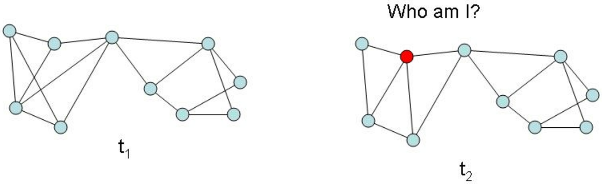
1. Setup as described previously.
2. Set $D = 0_{N \times N}$.
3. Set the first $n \times n$ block of D equal to the the dot product of the result of running the least squares Algorithm on B.
    3.1 While(Not Converged)
    3.2 $Y = g_d(A + D)$
    3.3 Set the unknown entries of D (such as those corresponding to $U_1 \times U_2$) to the dot products of the appropriate parts of Y.
4. Output$Y$

- Use the vectors to find the alias: closest named vector to the one associated with the alias.

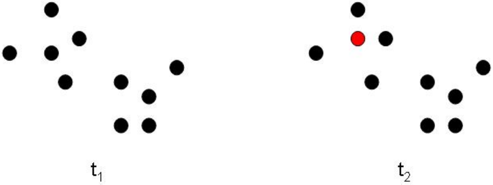
Alias Identification: k..allen→ phillip.allen



Cartoon



Social Space



Outline

Motivation

Definitions and Model

Alias Identification

Conclusions

Conclusions

- Social space provides a mechanism for modeling and inference on graphs and time series of graphs.
- Dot product graph model is simple, but easy to fit using linear algebra.
- Sparse matrix approaches can make this efficient:
    - There appears to be an $O(n^s)$, $2 < s < 3$ matrix multiply in the algorithm, in order to determine the stopping criterion (compute the error).
    - Some tricks can be played to reduce this for this application.
    - By using only the change in the diagonal for determining convergence, we eliminate the need for the full matrix multiply, replacing it with an $O(n)$ operation. Note that we only need to check the diagonal, since once this stops changing the algorithm produces a fixed point.
- It is possible to add covariates (measurements at the nodes) into the model and still use the linear algebra approach, but this work is preliminary.

Questions?

Contact Information: dmarchette@gmail.com